**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1**

**ПО ДИСЦИПЛИНЕ: «Информационные технологии и программирование»**

Выполнил:  
Ёркин Артём БВТ2208

Преподаватель:  
Харрасов К Р

г. Москва

2022

В программе intelJS используем встроенные функции для создания репозитория и

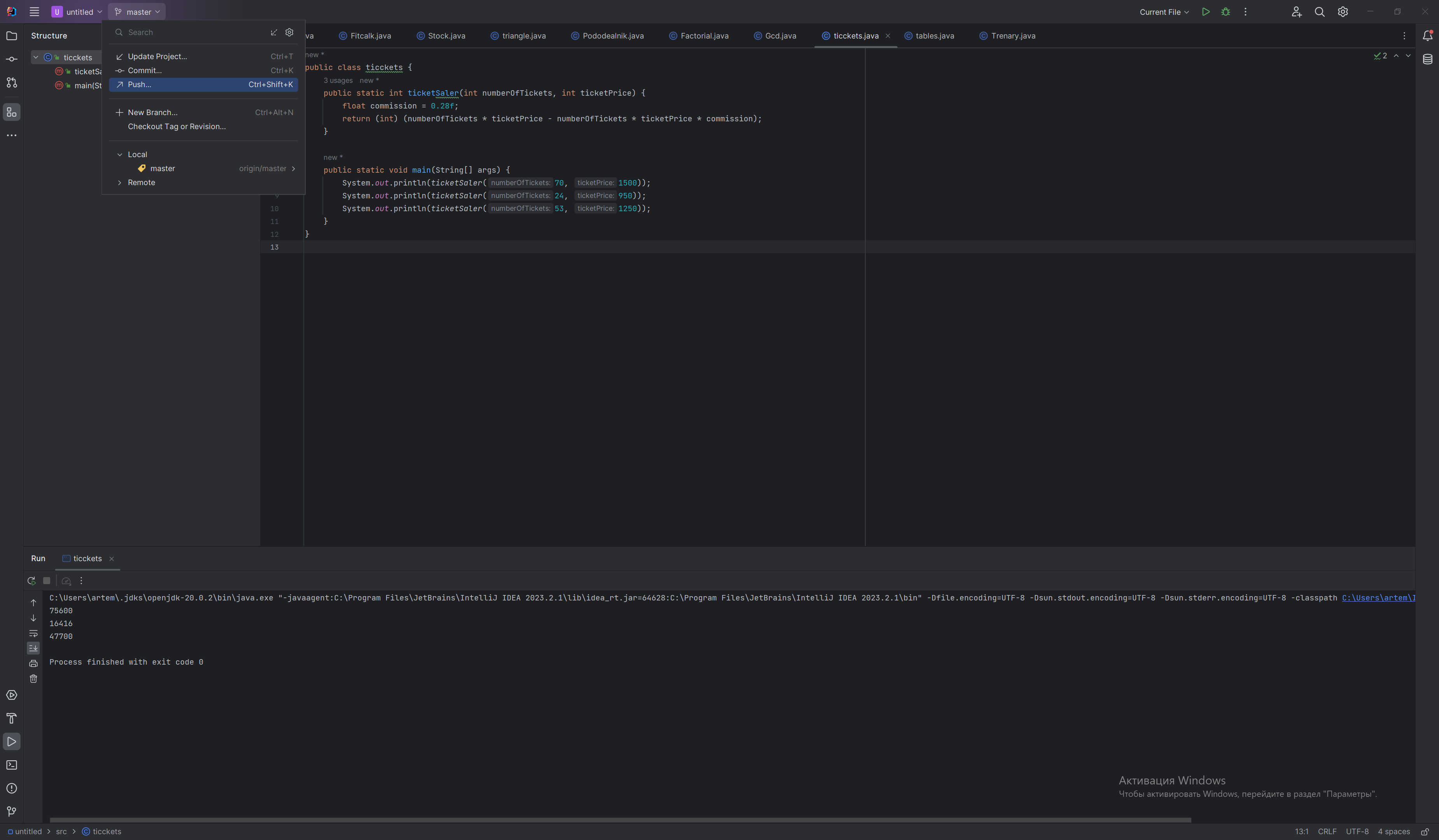


Рисунок1 – создание репозитория

Создаем некоторые изменения в файлах и делаем 2 коммита.

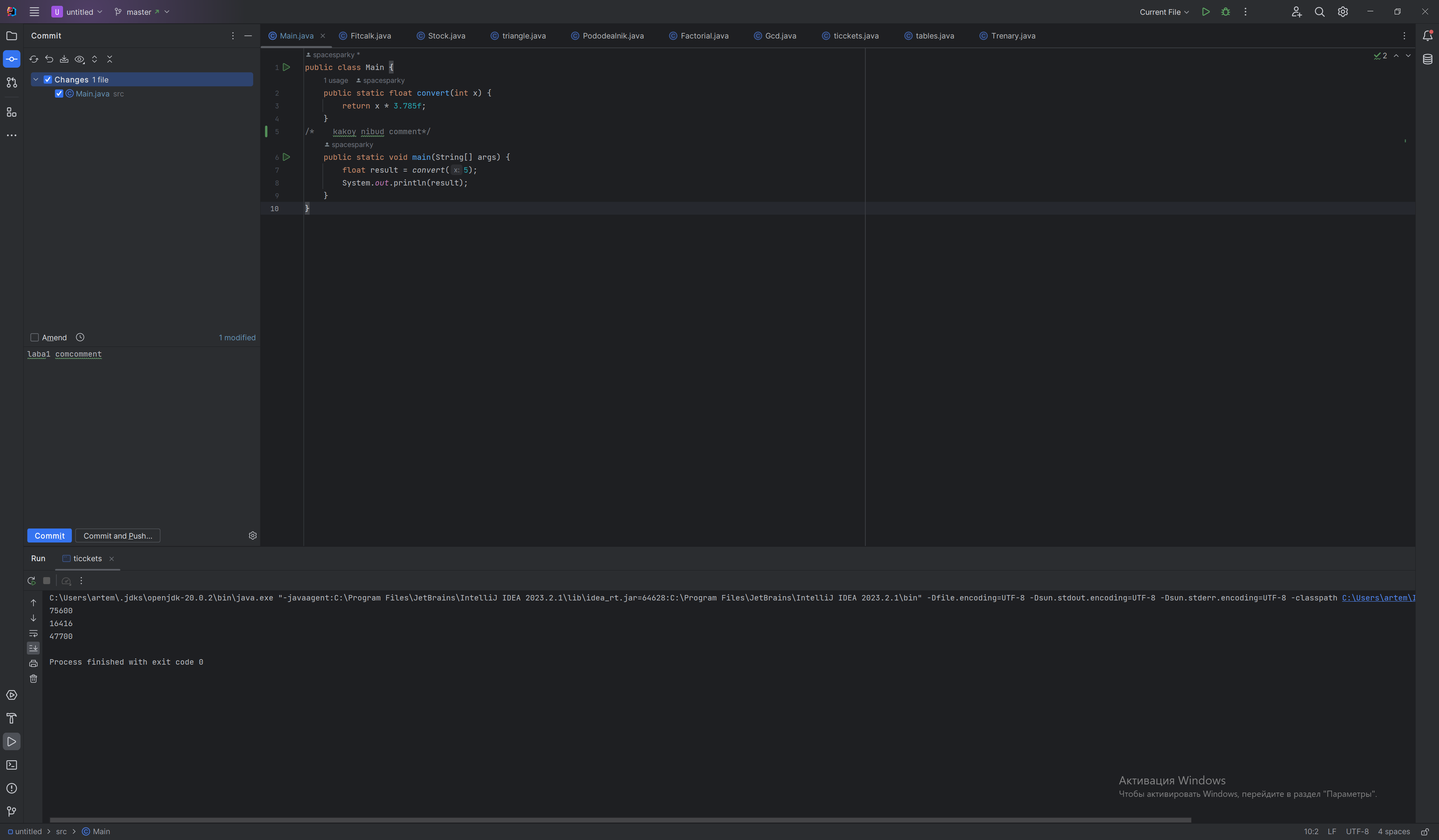


Рисунок2 – коммиты

Для отката изменений используем git revert HEAD и git checkout. Команда git revert просто создает коммит, отменяющий предыдущий.

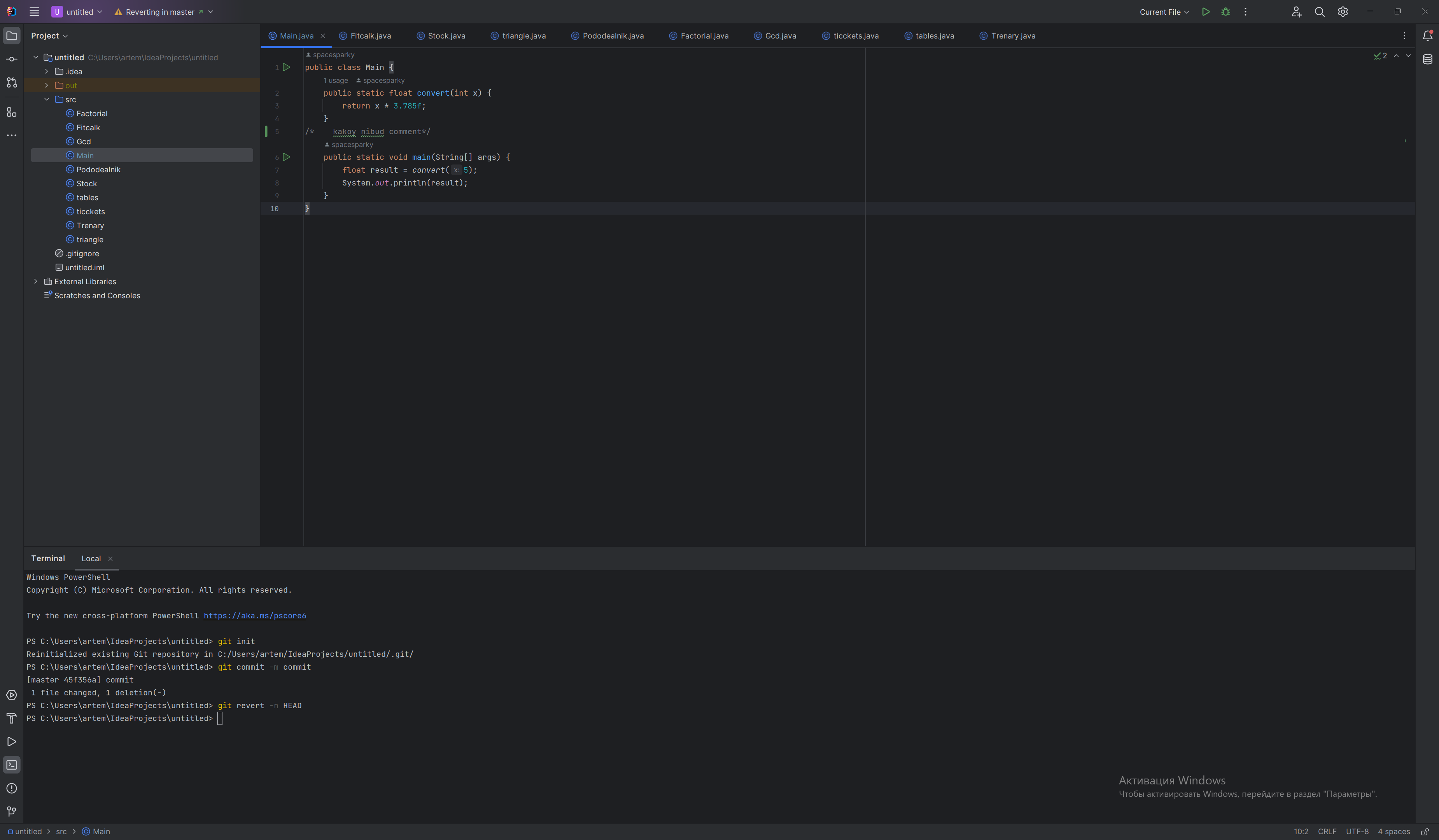


Рисунок3 – использование git revert

Команда git checkout переносит нас к коммиту, хэш которого мы смотрим через git log.

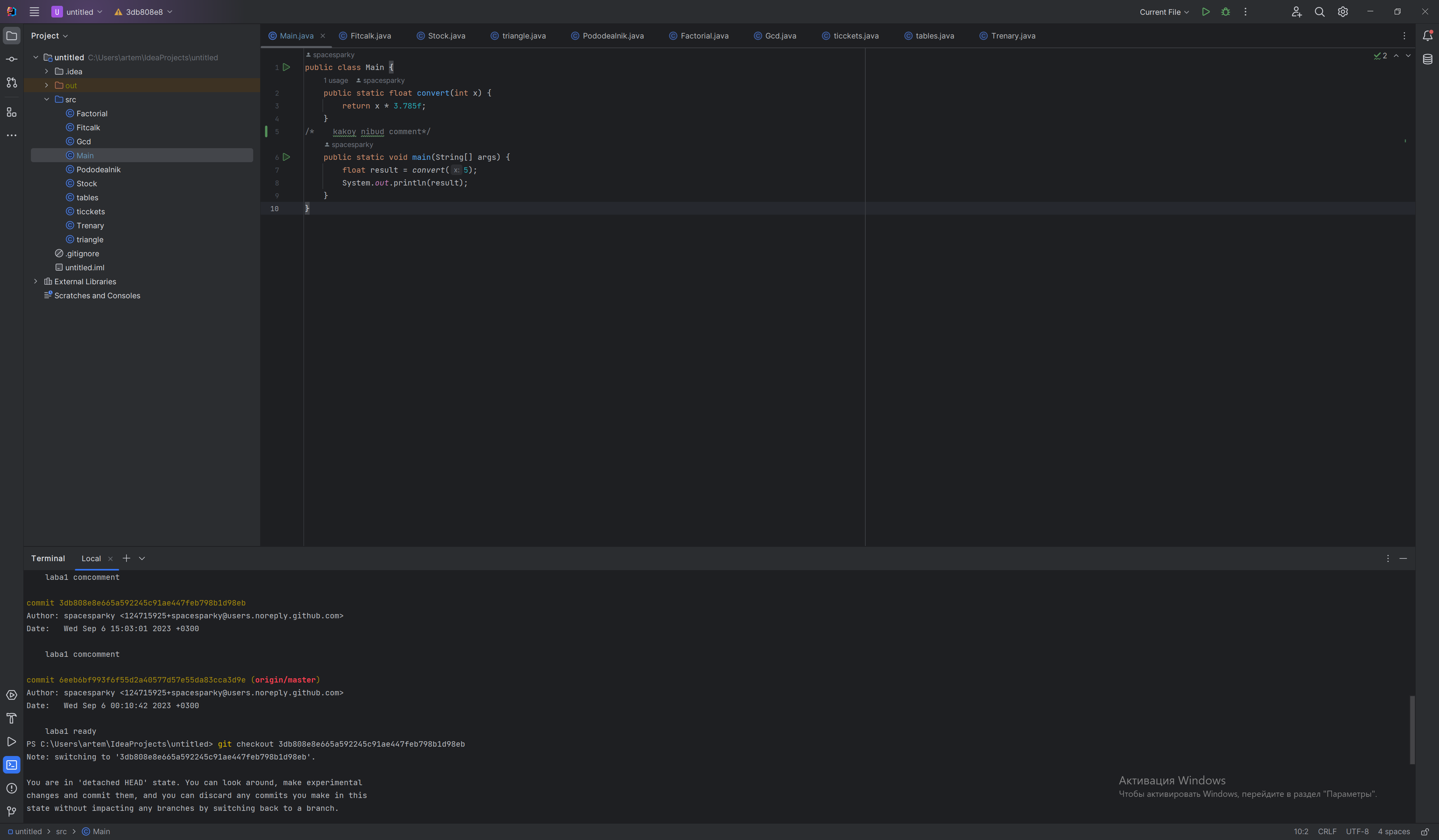


Рисунок4 – структура кода тестов и их успешное прохождение

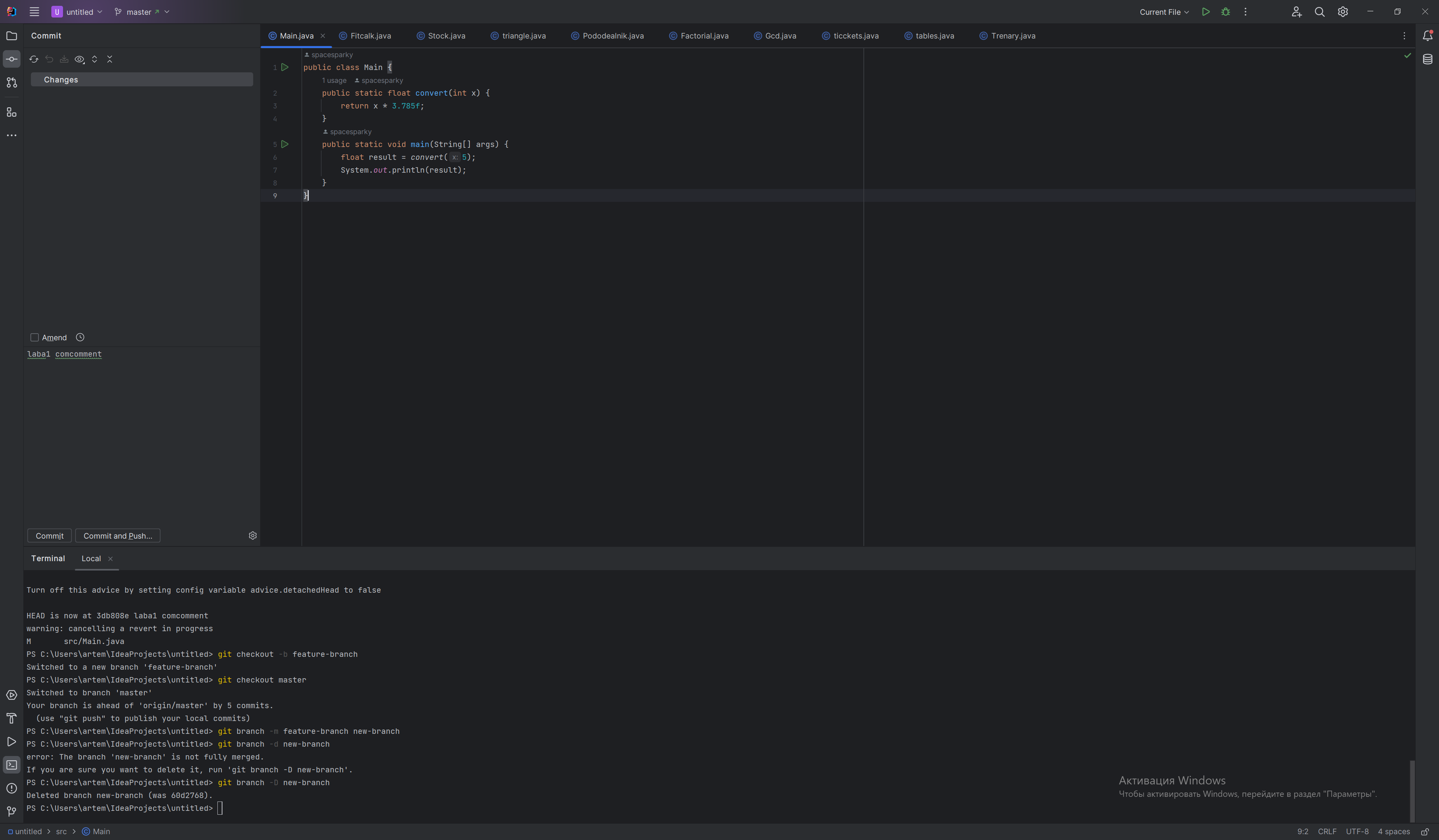


Рисунок5 Создание, переименование и удаление ветки

Для создания конфликта файлов нужно создать новую ветку и сделать на ней коммит с изменениями. Затем нужно перейти на ветку мастер и сделать команду git merge ‘название ветки’. После этого мы видим конфликт файлов, удаляем ненужную часть и делаем коммит.

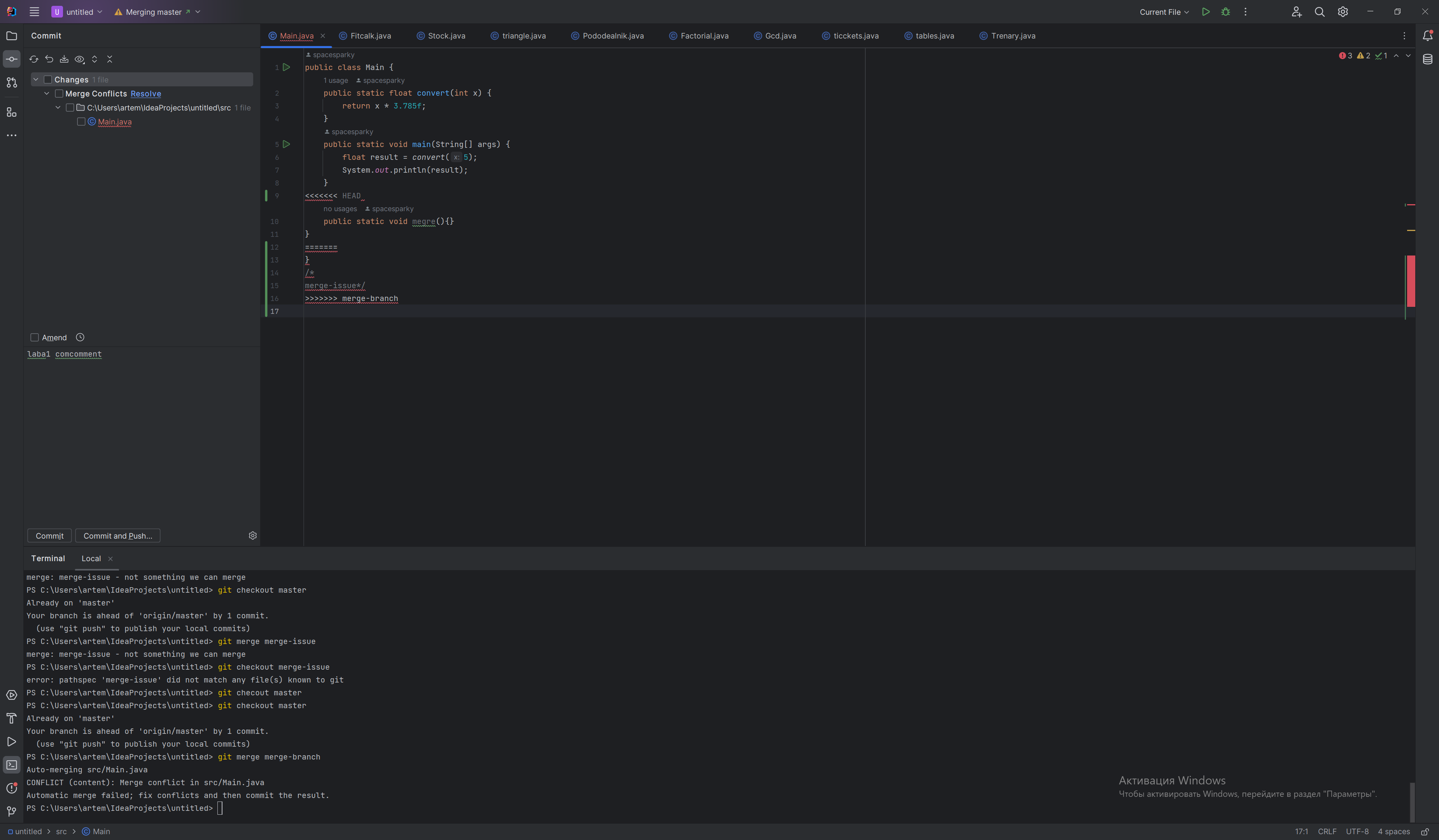


Рисунок6 – конфликт при слиянии веток

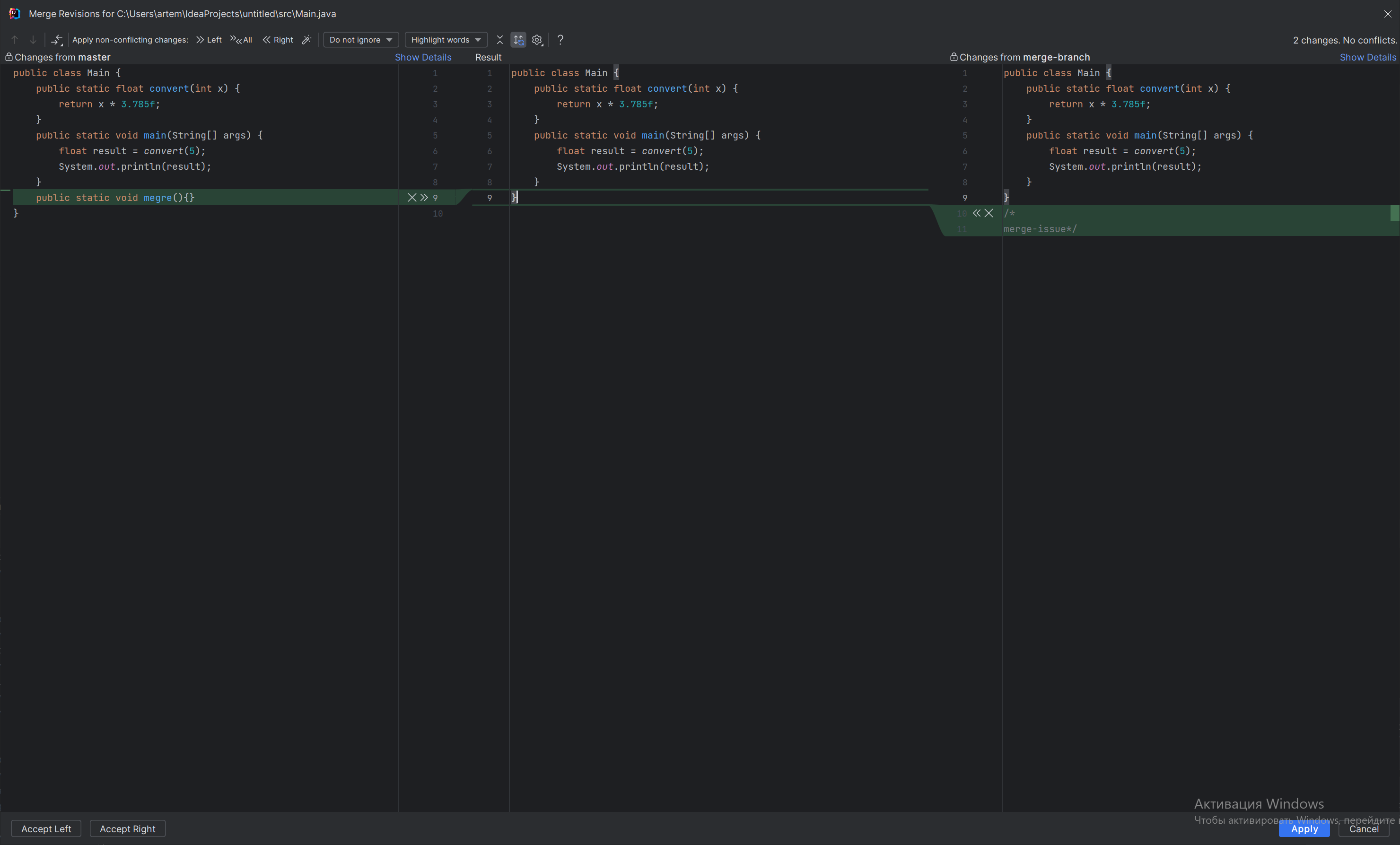


Рисунок7 – Разрешение конфликта утилитой inteljs

Среда разработки автоматически создает файл .gitignore для игнорирования файлов среды.

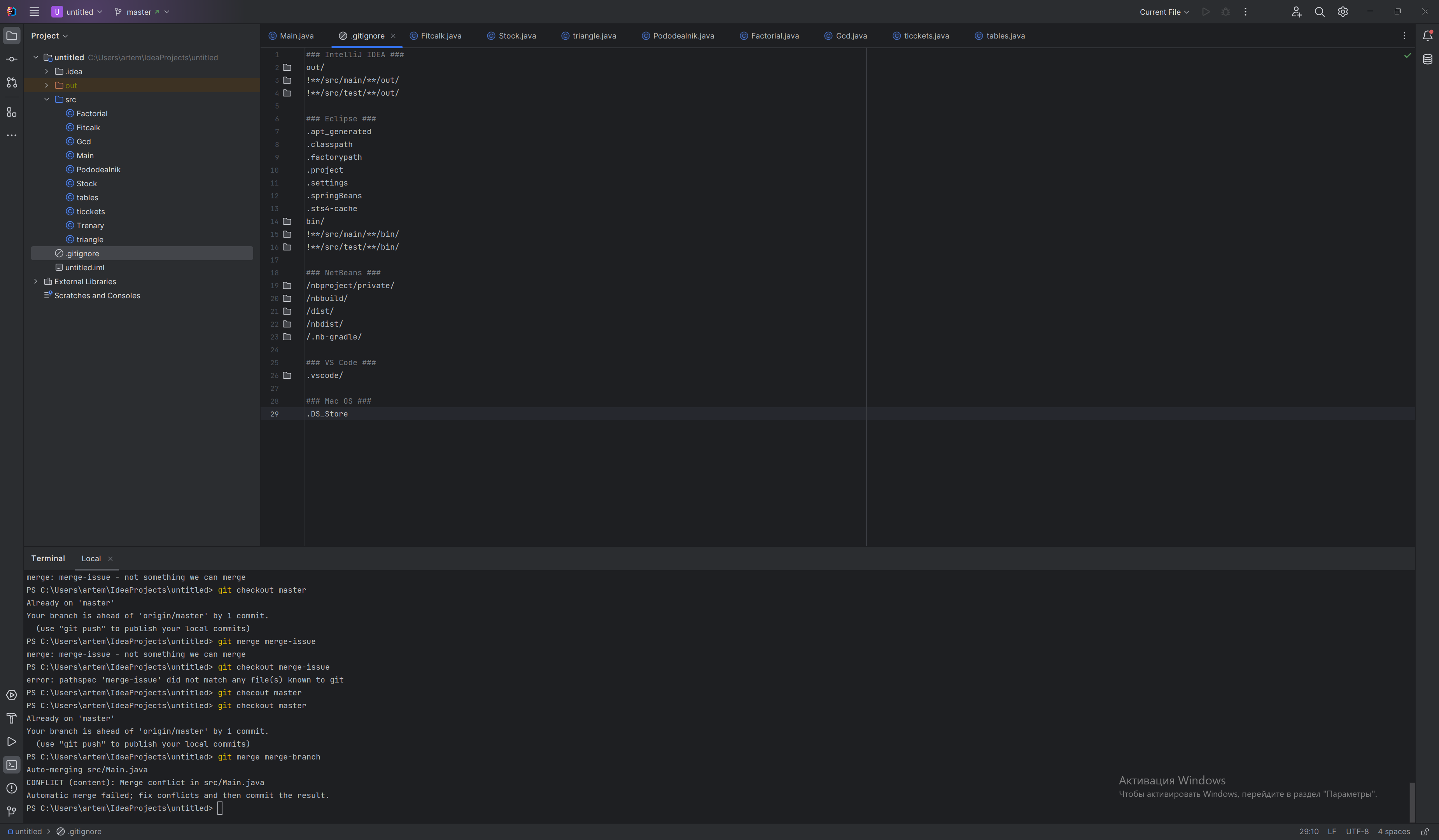


Рисунок8 – .gitignore

1. **\*** (звездочка): Сопоставляет любую последовательность символов, включая ноль символов. Например, **\*.txt** сопоставляется с любыми файлами, имеющими расширение ".txt".
2. **?** (вопросительный знак): Сопоставляет один любой символ. Например, **image?.jpg** сопоставляется с "image1.jpg", "image2.jpg" и так далее.
3. **[]** (квадратные скобки): Позволяют указать диапазон символов или набор символов для сопоставления. Например, **[aeiou]** сопоставляется с любой гласной буквой.
4. **{}** (фигурные скобки): Позволяют создавать списки альтернативных шаблонов. Например, **{\*.jpg,\*.png}** сопоставляется с файлами, имеющими расширение ".jpg" или ".png".
5. **!** (восклицательный знак): Исключает сопоставление файлов и директорий. Например, **!temp/\*.log** исключает все файлы с расширением ".log" в директории "temp".
6. **\*\*** (две звездочки): Рекурсивный поиск файлов и директорий внутри поддиректорий. Например, **src/\*\*/\*.java** сопоставляется со всеми файлами Java внутри директории "src" и ее поддиректорий.

Для вывода лога коммитов в измененном пользовательском формате в Git можно использовать команду **git log** с опцией **--pretty=format**. Эта опция позволяет вам настроить формат вывода лога коммитов с помощью строкового формата.

* **%h** - сокращенный хеш коммита.
* **%H** - полный хеш коммита.
* **%an** - имя автора коммита.
* **%ae** - email автора коммита.
* **%ad** - дата автора коммита (формат может быть настроен).
* **%s** - краткое описание коммита.
* **%B** - полное сообщение коммита.

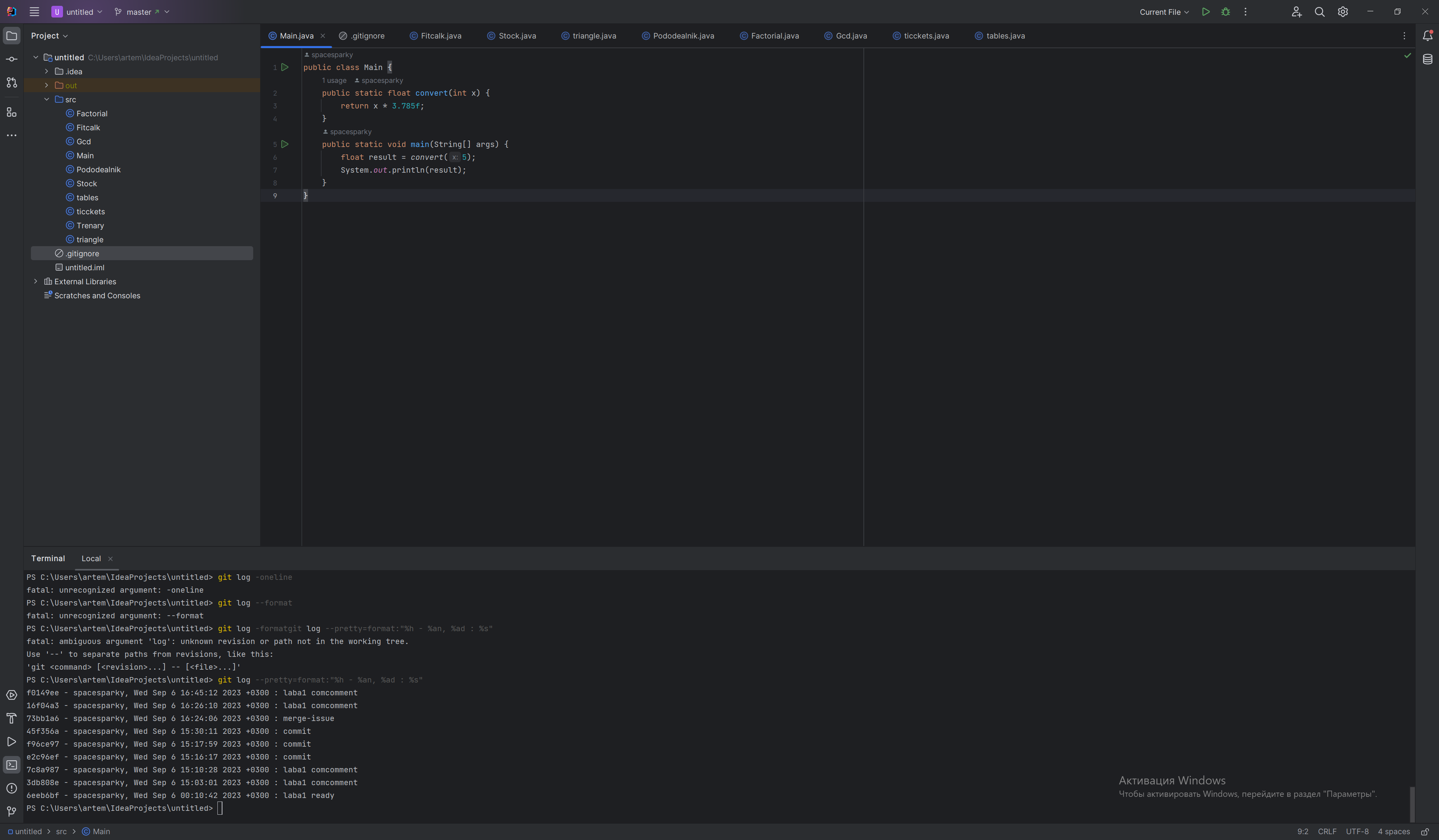


Рисунок9 – отформатированный gitlog

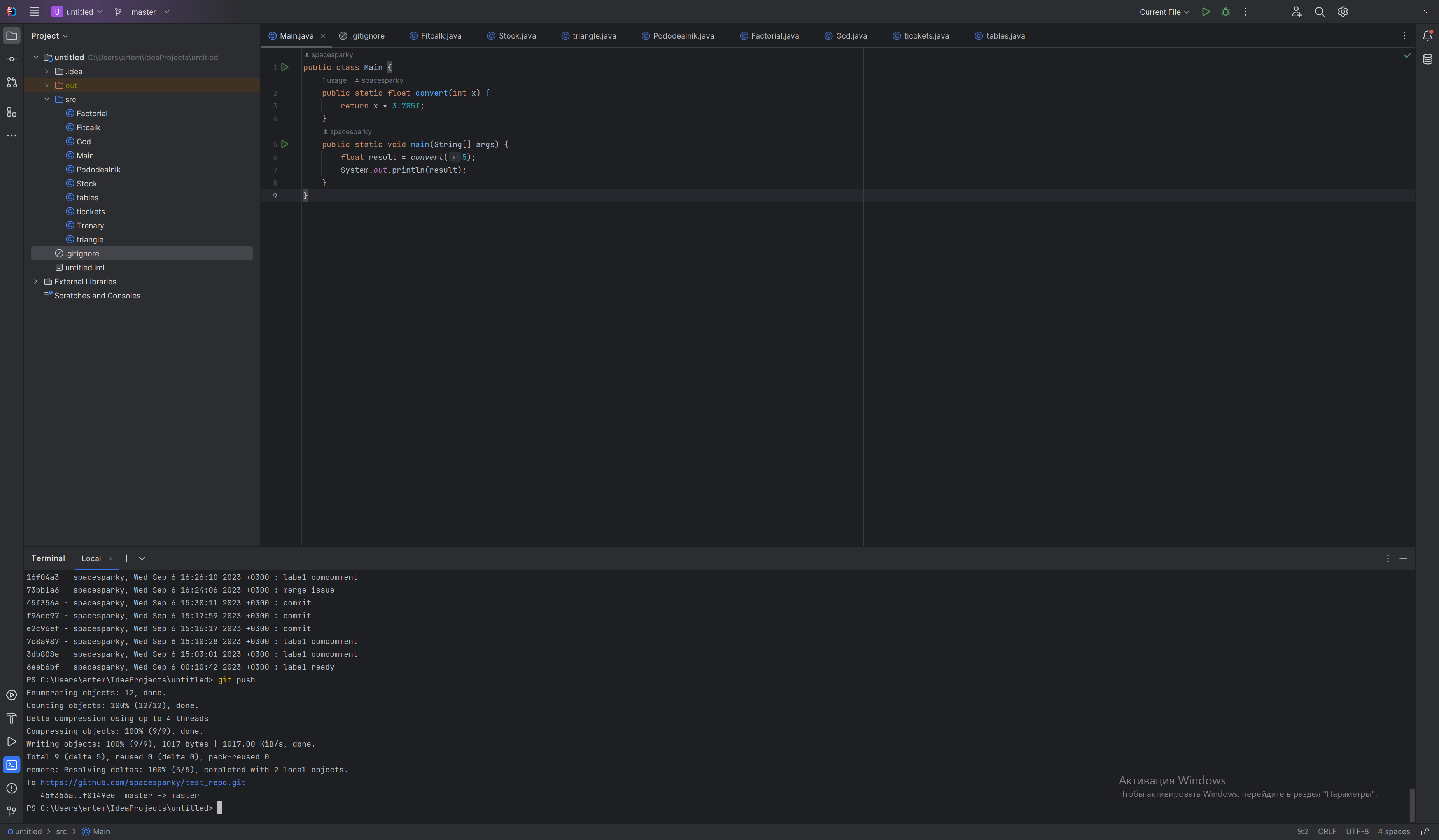


Рисунок10 – удаленный репозиторий на github

Для работы с чужим репозиторием используем команду git clone ‘url репозитория”. После клонирования чужого репозитория делаем коммиты и вводим url для пуша.

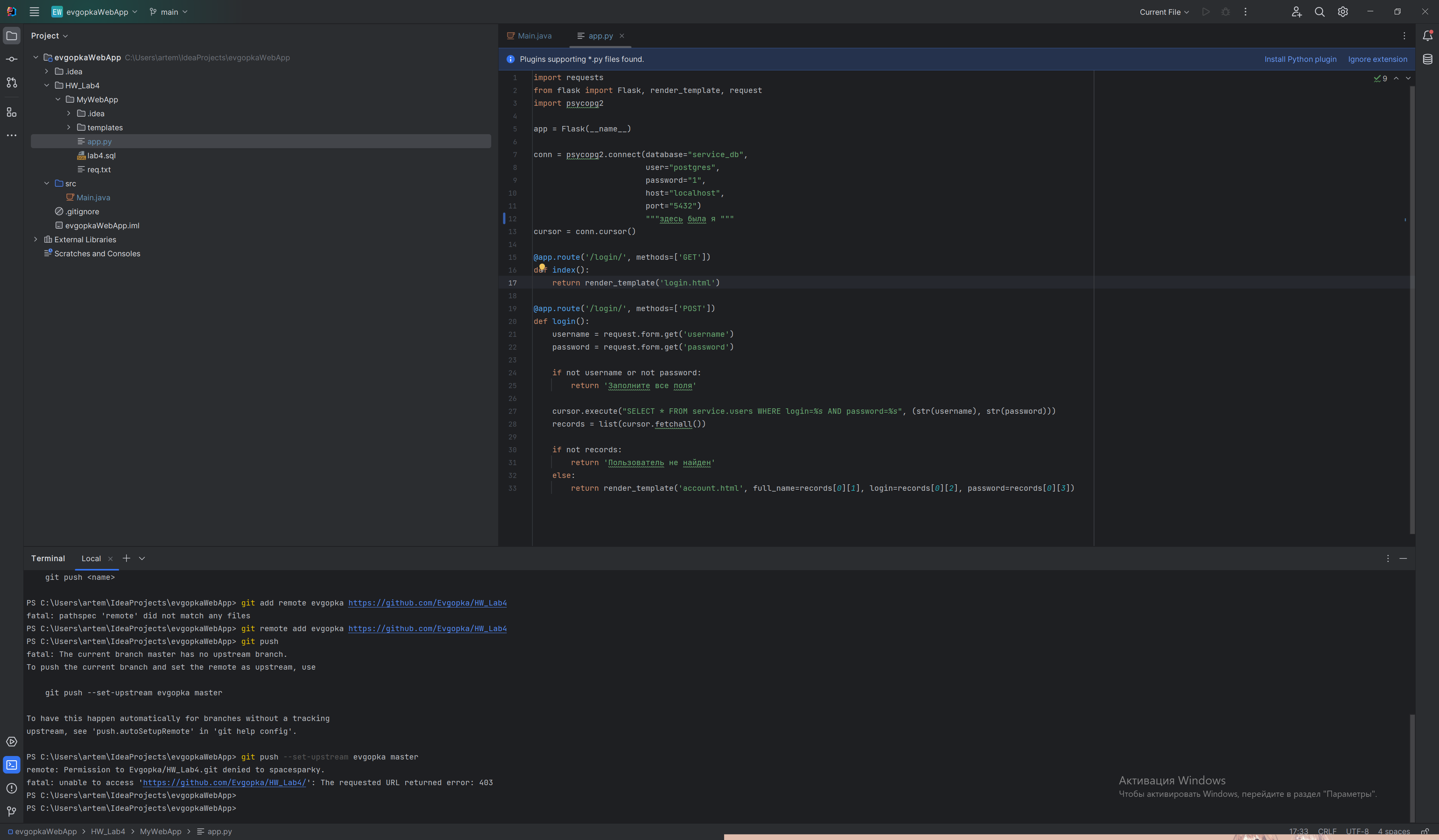


Рисунок10 – отказ в доступе на пуш в чужой пепозиторий

Чтобы пушить в чужие репозитории нужно сначала получить доступ к ним. К сожалению у меня нет таких репозиториев. Вот инсткрукция как его открыть:

Чтобы открыть доступ для других пользователей на внесение файлов (push) в свой репозиторий в системе контроля версий Git, вы можете использовать два основных способа:

1. **Добавление коллабораторов на GitHub (или другом хостинге Git):**

Если ваш репозиторий хранится на платформе, такой как GitHub, GitLab или Bitbucket, вы можете добавить коллабораторов, которые будут иметь право на внесение изменений в ваш репозиторий. Вот как это сделать на GitHub:

* + Перейдите на страницу вашего репозитория на GitHub.
  + Нажмите "Settings" (Настройки).
  + В меню слева выберите "Manage access" (Управление доступом).
  + Нажмите "Invite a collaborator" (Пригласить коллегу).
  + Введите имя пользователя или адрес электронной почты коллаборатора.
  + Нажмите "Add collaborator" (Добавить коллегу).

Коллаборатор получит уведомление и сможет вносить изменения в ваш репозиторий.

1. **Открытие репозитория для внешних контрибьюторов:**

Если вы хотите, чтобы кто-то извне мог предоставлять пулл-реквесты с изменениями, которые вы затем проверите и внесете в ваш репозиторий, вам необходимо настроить ваш репозиторий так, чтобы он принимал внешние пулл-реквесты. Это зависит от платформы, на которой вы храните ваш репозиторий, и требует настройки файлов **CONTRIBUTING.md** и **CODE\_OF\_CONDUCT.md**, а также возможно настроек безопасности.

Например, на GitHub вы можете настроить ваш репозиторий следующим образом:

* + Создайте файл **CONTRIBUTING.md** и опишите, какие правила и инструкции должны соблюдаться при предоставлении пулл-реквестов.
  + Создайте файл **CODE\_OF\_CONDUCT.md**, в котором описано, какой кодекс поведения должны соблюдать контрибьюторы.
  + Перейдите в раздел "Settings" (Настройки) вашего репозитория на GitHub.
  + В меню слева выберите "Options" (Настройки).
  + Прокрутите вниз до раздела "Features" (Функции).
  + Убедитесь, что установлена опция "Allow forking" (Разрешить клонирование).
  + Убедитесь, что опция "Restrict who can push to matching branches" (Ограничить, кто может пушить в соответствующие ветки) не установлена.

Вывод

Благодаря лабораторной были обновлены мои навыки работы с репозиториями github и системой гит.